# FEDD: Feature Extraction for Explicit Concept Drift Detection in Time Series

CAVALCANTE, R.; MINKU, L.L.; OLIVEIRA, A.

*DOI:*
10.1109/IJCNN.2016.7727274

# FEDD: Feature Extraction for Explicit Concept Drift Detection in Time Series

Rodolfo C. Cavalcante[1,3]
[1]Núcleo de Ciências Exatas
Universidade Federal de Alagoas
Arapiraca, Alagoas, Brasil
rodolfo.cavalcante@arapiraca.ufal.br

Leandro L. Minku[2]
[2]Department of Computer Science
University of Leicester
University Road, Leicester, UK
leandro.minku@leicester.ac.uk

Adriano L. I. Oliveira[3]
[3]Centro de Informática
Universidade Federal de Pernambuco
Recife, Pernambuco, Brasil
alio@cin.ufpe.br

***Abstract*—A time series is a sequence of observations collected over fixed sampling intervals. Several real-world dynamic processes can be modeled as a time series, such as stock price movements, exchange rates, temperatures, among others. As a special kind of data stream, a time series may present concept drift, which affects negatively time series analysis and forecasting. Explicit drift detection methods based on monitoring the time series features may provide a better understanding of how concepts evolve over time than methods based on monitoring the forecasting error of a base predictor. In this paper, we propose an online explicit drift detection method that identifies concept drifts in time series by monitoring time series features, called Feature Extraction for Explicit Concept Drift Detection (FEDD). Computational experiments showed that FEDD performed better than error-based approaches in several linear and nonlinear artificial time series with abrupt and gradual concept drifts.**

## I. Introduction

A time series is a sequence of observations collected over fixed sampling intervals [1]. Several dynamic processes can be modeled as time series, such as stock price movements, monthly sales of a company, the temperature of a city, exchange rates, among others. Time series forecasting can be considered one of the main challenges in the computational intelligence literature. It can be defined as follows. Let $S = \{x_1, ..., x_i, ...\}, x_i \in \mathbb{R}$, be a time series generated by a process $\mathcal{S}$. At each time stamp $t$, we want to predict the regression value $y = x_{t+n}$ based on a set of inputs $X = \{x_{t-p}, ..., x_t\}$, which are the last $p$ time series observations.

In the last decades, several approaches have been proposed for time series analysis and forecasting. Two major classes of these approaches are the traditional statistical models and the soft computing approaches [2]. Statistical models generally assume that the time series under study is generated from a linear process [3]. Soft computing techniques, on the other hand, are data-driven, self-adaptive methods able to capture nonlinear behavior of time series without statistical assumptions about the data [4].

Despite the fact that there is a vast literature on time series analysis, the majority of the existing approaches does not take into account that a time series is a special kind of data stream [5]. A data stream is a set of data observations which arrive sequentially item by item [6]. Dynamism is an inherent feature of data streams. This dynamism implies that patterns in a data stream may evolve over time and introduces a big challenge to traditional batch learning algorithms, that is the ability to permanently maintain an accurate decision model even in the presence of changes in the data stream. These changes are known as concept drifts [7].

Most of the approaches designed to time series analysis are unaware of concept drifts. These methods are based on the main assumption that time series concepts are stationary in such a way that the observations follow a fixed and immutable probability distribution. This assumption, however, may not hold for several industrial time series applications. For example, the time series of the sales of a product may change its behavior due to changes in government regulations or advertising campaigns. The time series of stock prices of a company may change its behavior due to changes in political and economical factors or due to changes in the investors psychology or expectations.

Concept drifts have been widely studied in classification problems [8]. The methods proposed for handling concept drifts can be divided into two main groups: (1) implicit or blinding methods and (2) explicit detection methods. Implicit methods [9], [10] are those that update the decision model in regular intervals, independently of the occurrence of concept drifts. The main issues of these approaches are the potential resource consumption to update the learned model even when the incoming data belong to the same concept and the potential overfitting of the learned model to the data.

Explicit drift detection methods [11], [12], [13], are those that monitor some statistics of the data stream in order to detect concept drifts. A statistical test is employed to detect changes in the observed data. An advantage of explicitly drift detection is that this approach works as a white box, by informing the user about the occurrence of concept drifts. Two common explicit drift detection approaches are those that monitor the error of a base-learner and those that monitor the data distribution features. The main issue of the error-based approaches is that the error level may not properly reflect concept drifts. Besides, these methods rely on the accuracy of the decision model used for prediction. If a poor training process is used to build the decision model, it may result in lots of false alarms or high miss-detection rates, due to generalization problems such as overfitting.

Concept drifts in time series forecasting have a key difference with respect to classification problems, requiring separate investigation and potentially different drift detection methods than for classification or other regression problems. Specifi-

cally, every change in $p(X)$ affects $p(y|X)$, since $p(X)$ and $p(y)$ are drawn from the same distribution. Due to this, we expect a drift detection method based on monitoring the data distribution to provide a better understanding of how concepts evolve over time than those based on monitoring the prediction error. Some approaches proposed in the literature investigated concept drift in time series in an explicit way by monitoring the data directly [14], [15], [16], [17]. However they detect changes in a retrospective way. Industrial applications typically demand real-time change detection methods, with minimum drift detection delay. Online drift detection methods for time series has been very little investigated in the literature.

In this paper, we propose an online explicit drift detection method that identifies concept drifts in time series by monitoring time series features. Our main research hypothesis is that some time series features can be used to define time series concepts. This hypothesis can be stated as follows: "by monitoring changes in time series features, it is possible to build an explicit drift detection method able to detect concept drift in an effective way, minimizing both drift detection delay, false alarms and miss-detection rates". The original contribution of this paper is FEDD (Feature Extraction for Explicit Concept Drift Detection), an online explicit drift detection method based on time series features. Experiments show that FEDD is effective in detecting drifts in linear and nonlinear time series with abrupt and gradual concept drifts.

The rest of this paper is organized as follows. Section II explains related work. Section III describes the proposed method in detail. Section IV presents the artificial data sets and the evaluation setup used in the study. Section V describes the computational experiments and discusses the results. Section VI concludes the paper and gives directions for further research.

## II. Related Work

In the literature, several researchers have investigated how to handle concept drift in classification problems. However in time series analysis, just a few researchers have attempted to solve this problem.

Guajardo et al. [10] proposed an implicit concept drift handling method for time series which is based on moving window and support vector regression (SVR). A moving window slides through the time series data stream in order to define the training and test sets. At each step the window moves, SVR is retrained with the portion of the window reserved for training and applied to the test set. The window size is adjusted to fit seasonal patterns of the time series and slides considering these cycles. An issue of this approach is that the seasonal patterns of a time series is typically not known a priori. Besides, several real time series have no well defined seasonal patterns, which prevents a widespread application of this method.

Gu et al. [18] used the similar idea of having a moving window for handling concept drift in time series implicitly by updating a base-predictor. In that work, the window has a variable size which is adjusted by a probabilistic model that defines which instances are in the retraining window. The probabilistic model gives more weight to more recent samples and to the samples more similar to the samples to be predicted. Samples with higher weights are used to fill the window.

Some explicit concept drift detection methods for time series have been proposed in the literature [14], [15], [19], [16], [17]. These methods have as main objective the detection of change points, which are the time instants when concept drifts have occurred. These approaches are based on a retrospective statistical analysis of the time series data or residuals for identifying changing points. An issue of these methods is the retrospective analysis, in which the detection is done just after receiving several samples from the data stream. Industrial applications typically require a real-time drift detection, since it allows the fast updating of the decision model and, consequently, the minimization of accuracy losses. Another issue of these methods is that they are designed to work just with abrupt concept drifts. In real-world applications, such as financial market, exchange rates and sales records, the gradual changes are more common than the abrupt ones.

Boracchi and Roveri [20] proposed an online concept drift detection for time series that present self-similarity. In the proposed approach, at each time instant, a data sequence of fixed size is extracted from the incoming data and the most similar previously seen data sequence is recovered from memory. A change indicator $x(t)$ is computed as the difference between the two sequences. When the arriving data sequence differs signficantly from previously seen sequences, the distribution of $x(t)$ changes, and a drift is detected. The intersection of confidence intervals (ICI) drift detection test [12] is used to detect changes in the distribution of $x(t)$. The main issue of this approach is that it is specifically designed to time series that present self-similarity and periodicity.

In a recent work [5], we investigated the use of online explicit drift detection in time series by means of monitoring the error of a regression model. In that work, we implemented the Drift Detection Mechanism (DDM) [11] and the Exponentially Weighted Moving Average for Concept Drift Detection (ECDD) [13] in combination with extreme learning machine (ELM) algorithm to build an adaptive prediction method. The issue of these approaches is that the drift detection is sensitive to problems in training the regression model, such as the parameter adjustment, overfitting and poor generalization, among others.

In this paper, we propose a novel online explicit drift detection method for time series that is able to deal with abrupt and gradual drifts by examining time series features. The novelty resides in monitoring the evolution of time series features to detect the occurrence of concept drifts.

## III. The FEDD Approach

In this section the proposed FEDD is described in detail. FEDD is an explicit drift detection method for time series which monitors some statistical features of the time series in order to identify changes in the underlying time series data distribution. The time series features are pre-defined descriptive statistics automatically calculated from the time series data. FEDD tests the occurrence of concept drifts by monitoring the evolution of these features.

FEDD has two main modules: the feature extraction (FE) module and the drift detection (DD) module. The FE module is responsible for extracting time series features. The DD module monitors the evolution of the time series features

along the process and tests the occurrence of concept drifts. Initially, a feature vector is extracted from the available time series observations. This initial vector summarizes the known concept. The DD module keeps a moving window that slides when a new sample is available. The features are recomputed on the current window and the feature vector on that window is compared with the initial feature vector. When these two feature vectors differ significantly, the DD module identifies it as a concept drift. Two distance metrics to compute the dissimilarities between these vectors were implemented and tested in FEDD. A statistical test is run over the distances in order to detect concept drifts.

FEDD is an online method since it is able to sequentially inspect incoming time series observations, one at time, in order to decide whether or not there is a change. Even though the sequential processing starts only after the initial feature vector is extracted, this is different from retrospective methods, where changes can only be detected *in a past fixed-length sequence that has already been received as a whole*.

### A. The FE Module

Several statistical features can be used to characterize a time series. In the literature, the analysis of statistical time series features has been used to solve important machine learning problems, such as time series classification [21], time series clustering [22], time series meta-learning [23], among others. Our main research hypothesis is that some time series features can be used to define time series concepts. In stationary conditions, the time series feature values are expected to be stationary. Whenever these features evolve over time, it can be interpreted as a concept drift.

Due to the nature of time series data, a wide range of changes may appear along the time series generation process. In order to capture these changes, the FE module computes 6 linear and 2 nonlinear time series features[1] to describe the time series concepts. The linear features are as follows.

1) The time series autocorrelation, which describes the similarity between observations in function of some lag [1]. The autocorrelations at the first five lags were used;
2) The time series partial autocorrelation, which describes the correlation that results after removing the effects of any correlations due to terms at shorter lags [1]. The partial autocorrelations at the first five lags were used;
3) The time series variance, which describes the degree of instability of the time series;
4) The skewness coefficient, which describes the asymmetry of the data around the sample mean;
5) The kurtosis coefficient, which describes how outlier-prone a data distribution is;
6) The turning points rate, which describes the degree of oscillation of the time series.

The nonlinear features attempt to describe the nonlinear behavior of a time series, which are common in real-world time series. The computed nonlinear features are as follows:

[1]Due to space restrictions, we do not include the features formulas.

1) The bicorrelation, also called three-point autocorrelation, which is a high order correlation feature that is described by the joint moment of three variables formed from the time series in terms of two delays $t$ and $t'$ [24]. The bicorrelations at the first three lags were used;
2) The mutual information, which is defined for two variables $\mathbf{X}$ and $\mathbf{Y}$ and is defined as the amount of information that is known of one variable when the other is given [24]. The mutual information at the first three lags were used.

With exception of the turning points rate, all features are computed for the differencing time series in order to soften the influence of trends.

### B. The DD Module

The DD module consists of two main components: (1) a distance function, which computes the dissimilarity among two feature vectors, and (2) a drift detection test, which tests the occurrence of significant changes in the distance level. In this work, we implemented and tested two distance metrics, namely the cosine distance and the Pearson distance [25]. These distance metrics were chosen because they are not influenced by the range of possible values each feature can assume. This is a requirement of the method, because the time series features have different scales. Since FEDD is designed to perform in an online fashion, it is not possible to use conventional normalization methods for rescaling the attributes, due to impossibility of finding maximum and minimum values, or the mean and standard deviations for the whole time series.

The cosine distance is the angular distance of two vectors while ignoring their scales. The cosine distance between two vectors $\mathbf{A} = \{a_1, ...a_d\}$ and $\mathbf{B} = \{b_1, ..., b_d\}$, of dimension $d$, is computed as follows:

$$dist_{cos}(\mathbf{A}, \mathbf{B}) = 1 - \frac{\langle \mathbf{A}, \mathbf{B} \rangle}{||\mathbf{A}||||\mathbf{B}||} = 1 - \frac{\sum_{i=1}^{d} a_i b_i}{\sqrt{\sum_{i=1}^{d} a_i^2} \sqrt{\sum_{i=1}^{d} b_i^2}}.$$

The Pearson distance is a dissimilarity metric based on Pearson's product-momentum correlation coefficient of two vectors. This metric describes the similarity in shape between the two vectors, and is computed as follows:

$$dist_{pear}(\mathbf{A}, \mathbf{B}) = 1 - Corr(\mathbf{A}, \mathbf{B}) = 1 - \frac{\sum_{i=1}^{d} (a_i - \overline{a})(b_i - \overline{b})}{\sqrt{\sum_{i=1}^{d} (a_i - \overline{a})^2} \sqrt{\sum_{i=1}^{d} (b_i - \overline{b})^2}}.$$

The drift detection test implemented in the DD module was the ECDD [13]. ECDD analyses the exponentially weighted moving average (EWMA) of a variable to identify changes in its values. EWMA is an estimator of the mean of a sequence of values of a variable which gives more importance to recent data, whereas older data is being progressively downweighted. Suppose a set of values for a variable $\{x_1, ..., x_n\}$ which presents a mean $\mu_0$ and standard deviation $\sigma_x$. The EWMA estimators for the variable are $Z_0 = \mu_0$ and $Z_t = (1 - \lambda)Z_{t-1} + \lambda x_t,\ t > 0$. The parameter $\lambda$ indicates the weight given to recent data when compared to

older data. The mean and standard deviations of $Zt$ are $\mu_t$ and $\sigma_{Z_t} = \sqrt{(\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})\sigma_x}$, respectively.

The DD module uses ECDD to monitor the distances between an initial feature vector and the current feature vector. When the whole time series data belong to the same context, it is expected the distances are stationary and $Z_t$ fluctuates around the distance mean $\mu_0$. However, when a change occurs, the distance stream presents a new mean $\mu_1$ and $Z_t$ now goes away from $\mu_0$ to $\mu_1$. In [13], two rules were defined to monitor concept drift based on EWMA. When $Zt > \mu_0 + W\sigma_{Z_t}$, a warning signal is triggered. When $Zt > \mu_0 + C\sigma_{Z_t}$, a change signal is triggered. The warning threshold and control limit, $W$ and $C$ respectively, are parameters of the method. The reason ECDD test was chosen to be used in FEDD is that it monitors the EWMA of the distances instead of monitoring the instantaneous distances or the simple average of the distances, being more robust to false alarms caused by noise and outliers in the time series. However, other drift tests could be integrated into FEDD instead of ECDD.

The steps of FEDD algorithm are detailed in Figure 1. The inputs of the algorithm are (Step 1): an initial subset of observations of a time series $\mathbf{X} = \{x_1, ..., xi, ..., x_m\}$, where $x_i \in \mathbb{R}$ is a time series sample, the window size $m$, the warning threshold $W$, and the drift threshold $C$. In Step 2, the initialization of some variables is done. The variable $s$ denotes the start of the known time series concept, $warn$ stores the instant when the first warning signal is triggered. The variable $below\_warn$ is used to define when the process can leave the warning level once it was reached.

Step 3 is repeated for every instant a new sample from the time series becomes available. If the difference between the current time step $t$ and the start of the current known concept is less than $m$, the algorithm does nothing, since the window is not yet filled. When $t-s$ is equal to the window size $m-1$ (Step 4), then FE computes the initial feature vector $fv_0$ for the time series samples in the moving window $\{x_s, ..., x_t\}$, where $s = 1$ for this initial vector (Step 5). The feature vector $fv_0$ is used as a reference feature vector for the drift test, since it represents the known concept.

When $t - s > m - 1$ (Step 6), the algorithm starts the online processing of the time series. In Step 7, the current feature vector $fv_t$ is firstly computed by the FE module for the instances in the moving window. In Step 8, the distance between the initial feature vector and the current feature vector in time $t$ ($d_t$) is computed using one of the chosen distances explained before. In Step 9, the algorithm computes the distance average ($\mu_d$), the EWMA estimator of the distances ($Z_t^d$) and the standard deviation of $Z_t^d$, denoted by $\sigma_{Z_t^d}$. These are the statistics used in the concept drift test.

In Steps 10 and 14, the DD module uses the statistics of the sequential distances to identify if a warning level and/or a drift level was reached, respectively. If the warning signal was never reached and $Z_t^d$ is above the warning level (Step 10), a warning signal is triggered (Step 11) and the instant $t$ is kept in memory, since it marks the potential beginning of a new concept (Step 12). If the drift level is reached (Step 14), then a drift signal is triggered (Step 15), the variable $s$, which marks the beginning of the known concept, is updated with the instant which the warning level was reached, and the other control variables are reset (Step 16). The process is then reinitialized.

```
1: Inputs: X = {x_1, ..., x_m}, m, W, C.
2: s = 1, warn = 0, below_warn = 0
3: for (each instant t a new instance x_t arrives) do
4:     if (t − s = m − 1) then
5:         fv_0 = FE({x_s, ..., x_t})
6:     else if (t − s > m − 1) then
7:         fv_t = FE({x_{t−m+1}..., x_t})
8:         d_t = dist(fv_0, fv_t)
9:         compute μ_d, Z_t^d and σ_{Z_t^d}
10:        if ((warn = 0) and (Z_t^d > (μ_d + W ∗ σ_{Z_t^d}))) then
11:            trigger a warning signal
12:            warn = t
13:        end if
14:        if (Z_t^d > (μ_d + C ∗ σ_{Z_t^d})) then
15:            trigger a drift signal
16:            s = warn, warn = 0, below_warn = 0
17:        end if
18:        if (warn > 0) then
19:            if (Z_t^d > (μ_d + W ∗ σ_{Z_t^d}))) then
20:                below_warn = below_warn + 1
21:            end if
22:            if (below_warn = 10) then
23:                warn = 0
24:                below_warn = 0
25:            end if
26:        end if
27:    end if
28: end for
```

Fig. 1. The FEDD algorithm.

The FEDD algorithm still defines a mechanism that allows the process leaving the warning level (Steps 18–26). Once the warning level is reached, if $Z_t^d$ is below the warning threshold for 10 time steps, the process returns to the normal state. The value 10 was empirically defined and not optimized. This mechanism helps to prevent instances of the old concept to be included in the definition of a new concept when a drift is detected.

## IV. Computational Experiments

In this section we describe the metrics used to evaluate the methods, the data sets and the evaluation setup used in the experiments.

The main objective of the experiments with FEDD is to validate our hypothesis that by monitoring features of a time series we can improve the concept drift detection in comparison with error-based explicit drift detection methods. In order to do so, we compare FEDD with the ELM_ECDD method proposed in [5], since it is based on prediction error and uses the same drift test as FEDD. The second objective is to verify the drift identification accuracy of FEDD in comparison with other error-based drift detection methods. So, we compare FEDD with ELM_DDM and ELM_PHt. DDM and the Page-Hinkley test (PHt) [26] are promising drift detection tests proposed in the literature. The choice of ELM as a base-regressor is due to the fact that ELM has been widely used in regression and time series forecasting with good generalization performance, besides presenting a very fast training [27]. The third objective of the experiment is to evaluate the use of cosine and Pearson distance functions in FEDD.

The performance metrics used to evaluate the drift detection accuracy of the compared methods are: (1) the number of false alarms, (2) the miss-detection rates and (3) the delay of detection. A false alarm is a type-I error, which consists in a false positive drift identification. A miss-detection is a failure in detecting a drift when it actually happens in data stream, which is a type-II error. The miss-detection rate is the ratio between the number of miss-detections and the number of known concept drifts in the time series, and is given in percentage. The drift detection delay is the amount of instances the algorithm needed until the test can detect the occurrence of a known drift. We compute it as the sum of the delays presented by the method considering all drifts that exist in the time series. When a miss-detection occurs, all the time series observations that belong to the new concept are counted as the delay of the method. It is worth noting that, despite the fact that miss-detection and detection delay are related metrics, they describe different aspects of the detection process.

### A. Data Set Description

Despite the fact that concept drift is not a new research area, the effects of concept drift in time series are not widely studied. There is a lack of appropriate data sets aimed for studies of concept drift in regression tasks as well as in time series analysis. By working with real-world data sets, it is not possible to know exactly when a drift effectively occurs, the kind of drift or even if there is a drift on the data [28]. Artificial data sets, on the other hand, allow an effective analysis of the drift detection performance.

In this work, we create artificial data sets[2] which comprise linear and nonlinear time series affected by abrupt and gradual concept drifts. The simulated time series have size of 12.000 data points. Linear time series were simulated by autoregressive processes (AR) to generate the time series points. A time series $\mathbf{X}$ is an autoregressive process of order $p$, abbreviated as $AR(p)$, if its points can be defined as $x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + ... + \alpha_p x_{t-p} + w_t$, where $\alpha 1, ..., \alpha_p$ are the model parameters and $\{w_t : t = 1, ..., n\}$ is a Gaussian white noise time series where the variables $w_1, ..., w_n$ are independent and identically distributed and follow a normal distribution ($w_t \sim N(0, \sigma^2)$) [1].

The artificial linear time series data set is composed by 120 time series, which can be divided into three groups: (1) AR(4) time series affected by changes in the AR parameters and in $\sigma^2$, (2) AR(6) time series affected by changes in the AR parameters and in $\sigma^2$ and (3) AR($p$) time series affected by changes in the order $p$, in the AR parameters and in $\sigma^2$. These time series are affected by 3 drifts. In each group, half of the time series are affected by abrupt concept drifts and half is affected by gradual drifts. Table I reports the AR parameters for each group of linear time series. Abrupt drifts were simulated by an instantaneous change in the parameters. Gradual drifts were simulated by inserting a gradual change in the parameters. The size of a gradual change corresponds to 10% of the data points of the new concept.

We also simulate nonlinear time series to perform the evaluation of the proposed methods. The nonlinear time series models used to create this data set were proposed in [29].

[2]These data sets will be available for download.

TABLE I. LINEAR TIME SERIES DATA SET DESCRIPTION.

| TS Group | Concept | $\alpha$ | $\sigma^2$ |
|---|---|---|---|
| Linear 1 | 1 | $\{0.9, -0.2, 0.8, -0.5\}$ | 0.5 |
| | 2 | $\{-0.3, 1.4, 0.4, -0.5\}$ | 1.5 |
| | 3 | $\{1.5, -0.4, -0.3, 0.2\}$ | 2.5 |
| | 4 | $\{-0.1, 1.4, 0.4, -0.7\}$ | 3.5 |
| Linear 2 | 1 | $\{1.1, -0.6, 0.8, -0.5, -0.1, 0.3\}$ | 0.5 |
| | 2 | $\{-0.1, 1.2, 0.4, 0.3, -0.2, -0.6\}$ | 1.5 |
| | 3 | $\{1.2, -0.4, -0.3, 0.7, -0.6, 0.4\}$ | 2.5 |
| | 4 | $\{-0.1, 1.1, 0.5, 0.2, -0.2, -0.5\}$ | 3.5 |
| Linear 3 | 1 | $\{0.5, 0.5\}$ | 0.5 |
| | 2 | $\{1.5, 0.5\}$ | 1.5 |
| | 3 | $\{0.9, -0.2, 0.8, -0.5\}$ | 2.5 |
| | 4 | $\{0.9, 0.8, -0.6, 0.2, -0.5, -0.2, 0.4\}$ | 3.5 |

This data set is also composed by 120 time series with 3 drifts, divided into three groups: (1) nonlinear moving average model (eq. 1), (2) smooth transition autoregressive model 1 (eq. 2) and (3) smooth transition autoregressive model 2. (eq. 3). The concept drifts were simulated by changing the parameters of the models and $\sigma^2$, similarly to the linear time series data set. In each group, half of the number of time series are affected by abrupt concept drifts and half is affected by gradual drifts. The parameters used to generate these time series are given in Table II. The simulation of the abrupt and gradual drifts were performed in a similar way to the linear time series.

$$x_t = w_t - \alpha_1 w(t-1) + \alpha_2 w(t-2) + \alpha_3 w(t-1)w(t-2) - \alpha_4 w(t-2)^2 \quad (1)$$

$$x_t = [\alpha_1 x(t-1) + \alpha_2 x(t-2) + \alpha_3 x(t-3) + \alpha_4 x(t-4)] * \\ * [1 - exp(-10x(t-1)]^{-1} + w_t \quad (2)$$

$$x_t = \alpha_1 x(t-1) + \alpha_2 x(t-2) + [\alpha_3 x(t-1) + \alpha_1 x(t-2)] * \\ * [1 - exp(-10x(t-1)]^{-1} + w_t \quad (3)$$

TABLE II. NONLINEAR TIME SERIES DATA SET DESCRIPTION.

| TS Group | Concept | $\alpha$ | $\sigma^2$ |
|---|---|---|---|
| Non-Linear 1 | 1 | $\{0.9, -0.2, 0.8, -0.5\}$ | 0.5 |
| | 2 | $\{-0.3, 1.4, 0.4, -0.5\}$ | 1.5 |
| | 3 | $\{1.5, -0.4, -0.3, 0.2\}$ | 2.5 |
| | 4 | $\{-0.1, 1.4, 0.4, -0.7\}$ | 3.5 |
| Non-Linear 2 | 1 | $\{0.9, -0.2, 0.8, -0.5\}$ | 0.5 |
| | 2 | $\{-0.3, 1.4, 0.4, -0.5\}$ | 1.5 |
| | 3 | $\{1.5, -0.4, -0.3, 0.2\}$ | 2.5 |
| | 4 | $\{-0.1, 1.4, 0.4, -0.7\}$ | 3.5 |
| Non-Linear 3 | 1 | $\{-0.5; 0.8; -0.2; 0.9\}$ | 0.5 |
| | 2 | $\{-0.5; 0.4; 1.4; -0.3\}$ | 1.5 |
| | 3 | $\{0.2; -0.3; -0.4; 1.5\}$ | 2.5 |
| | 4 | $\{-0.7; 0.4; 1.4; -0.1\}$ | 3.5 |

One example of each group of linear and nonlinear time series with abrupt concept drifts is illustrated in Figure 2. The vertical bars indicate the concept drifts.

### B. Evaluation Setup

In the experiments, the compared methods were performed in the 240 generated time series. For each time series, the ELM-based drift detectors were run 30 times and the averaged results were used in the comparison. Since FEDD with cosine distance (FEDD$_{cos}$) and FEDD with Pearson distance (FEDD$_{pear}$) are deterministic, only one run of these methods

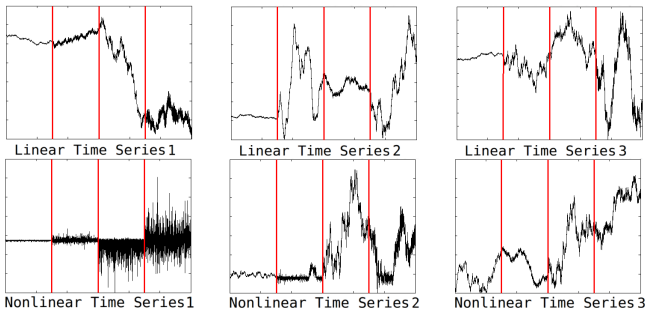


Fig. 2. Linear time series (above) and nonlinear time series (below) with abrupt concept drifts.

in each time series was performed. In order to assess the statistical significance of the results, we use the Friedman non-parametric test [30], with $\alpha = 0.05$ confidence level, according to the approach proposed in [31]. This evaluation approach allows the simultaneous comparison of several methods considering several different data sets. The null hypothesis is that there is no significant differences between the results. If the null hypothesis is rejected, the Nemenyi *post hoc* test [32] with 95% confidence is used identify the best results.

We initially performed experiments to identify the best parameter settings for each method. We consider the best set of parameters for a method the one that minimizes the sum of false alarms and miss-detections, since they are drift identification errors. In case of ties, the parameter set which provides the lowest drift detection delay is chosen.

FEDD$_{cos}$ and FEDD$_{pear}$ parameters are the window size ($m$), the weight given to the more recent data ($\lambda$) in calculating the EWMA, the warning threshold ($W$) and the drift threshold ($C$) of the ECDD drift test. We set the weight as $\lambda = 0.2$, since higher values for $\lambda$ give more importance to more recent values, which may make the test more sensitive to noise and outliers. We performed preliminary experiments using the following values for the parameters: $m \in \{100, 200, 300\}$, $W \in \{0.5, 1.0, 2.0\}$ and $C \in \{1.0, 1.5, 2.0, 3.0\}$. The best parameter set found was $m = 300$, $W = 1.0$ and $C = 1.5$.

The methods based on prediction error for detecting concept drift (ELM_DDM, ELM_ECDD and ELM_PHt) use ELM with a sigmoidal function as activation function and have as parameter the number of hidden neurons ($h$). Tests were performed with $h \in \{10, 20\}$ and the preliminary results indicated better accuracy with $h = 10$. The initial training set for ELM was set to 1000 time series data samples.

ELM_DDM has as parameters the warning threshold ($W$) and the drift threshold ($C$). We also established as parameter of this method the minimum number of instances ($n$) to retrain ELM after a drift detection as a parameter of the ELM_DDM method. Experiments were performed using the following values for the parameters: $n \in \{100, 200, 300, 400\}$, $W \in \{0.5, 1.0, 2.0\}$ and $C \in \{1.0, 1.5, 2.0, 3.0\}$. The preliminary experiments indicated that the parameter set with best results was $n = 400$, $W = 2.0$ and $C = 3.0$.

ELM_ECDD has the same parameters defined for ECDD ($\lambda$, $W$ and $C$), and the minimum number of instances ($n$) to retrain ELM after a drift. We fixed $\lambda = 0.2$, as in FEDD. Preliminary experiments were performed with the following values for the parameters: $n \in \{100, 200, 300, 400\}$, $W \in \{0.5, 1.0, 2.0\}$ and $C \in \{1.0, 1.5, 2.0, 3.0\}$. The best parameter set found was $n = 400$, $W = 1.0$ and $C = 1.5$.

ELM_PHt has as parameters $W$, $C$ and $n$, and an additional parameter, the discount factor $\delta$ used to compute the cumulative error used as the statistic of the PHt test. In preliminary experiments we evaluate $W$, $C$ and $n$ with the same set of values as in experiments with ELM_DDM and ELM_ECDD. We try $\delta \in \{0.01, 0.05, 0.10\}$. The best parameter set found for ELM_PHt was $n = 400$, $\delta = 0.05$, $W = 1.0$ and $C = 2.0$.

## V. Experimental Results and Discussion

The first objective of the experiments is to compare FEDD with ELM_ECDD, in order to validate our hypothesis that analyzing the time series features can improve online concept drift detection in comparison to using the error of a base predictor. Figure 3 presents the Friedman ranks with the Nemenyi critical difference for the three metrics evaluated. Methods that are significantly different (at $p = 0,05$) have ranks which differ by at least the critical difference.

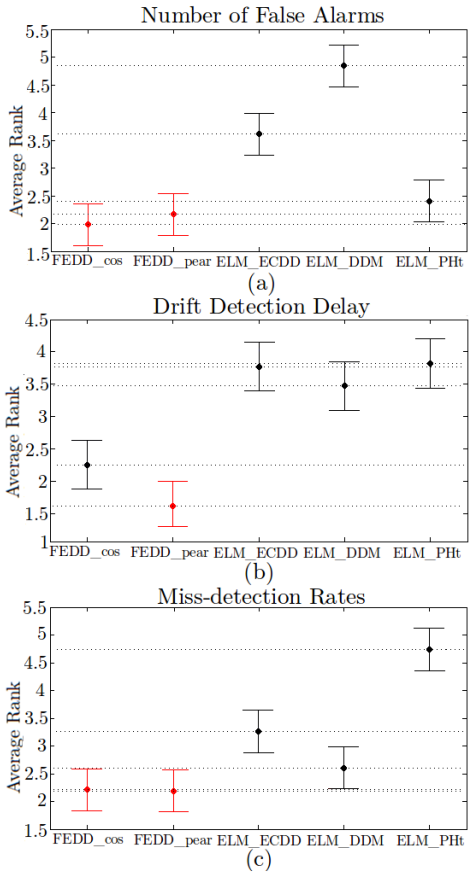


Fig. 3. Nemenyi test for all metrics. (a) Number of false alarms. (b) Drift detection delay. (c) Miss-detection rates.

According to the Friedman ranks and associated Nemenyi critical difference, FEDD_*cos* and FEDD_*pear* outperform ELM_ECDD in terms of false alarms (Figure 3(a)), drift detection delay (Figure 3(b)) and miss-detection rates (Figure 3(c)). Therefore, we can conclude that the monitoring of

feature vectors helped FEDD to significantly improve concept drift detection in comparison to monitoring ELM prediction error.

Figure 4 further illustrates the magnitudes of the differences in performance. In terms of false alarms (Figure 4(a)), one can see that in time series with abrupt drifts, in 50% of the cases, the FEDD_$cos$ and FEDD_$pear$ presented 0 false alarms. In the time series with gradual drifts, FEDD_$cos$, FEDD_$pear$ presented 0 false alarms in almost all time series, which configures the best results. The ELM_ECDD method presented similar behavior in both time series with abrupt and gradual drifts.

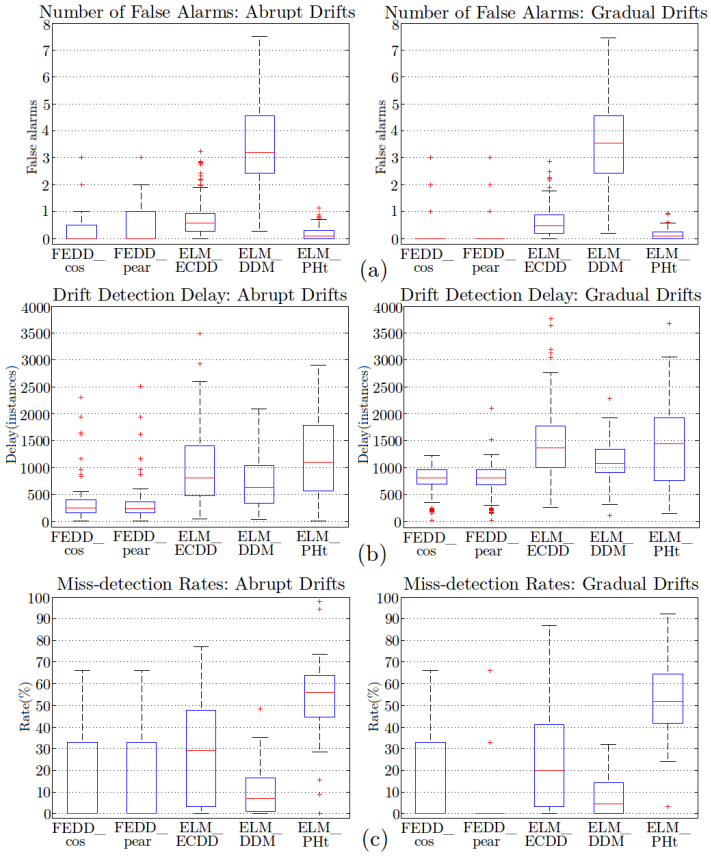


Fig. 4. Comparison of all methods in time series with abrupt (left) and gradual (right) concept drifts. (a) Number of false alarms. (b) Drift detection delay. (c) Miss detection rates.

Figure 4(b) shows the box-plots of results in terms of drift detection delay. As one can see, both FEDD_$cos$, FEDD_$pear$ and ELM_ECDD have the drift detection delay increased in time series with gradual concept drifts. The box-plots show that in time series with gradual concept drifts, in at about 75% of the tested time series, the FEDD approaches needed less that 1000 instances before detecting a concept drift. The ELM_ECDD, on the other hand, needed to receive at least 1000 instances before detecting a drift in more than 75% of the time series with gradual drifts.

Figure 4(c) allows a comparison of the methods in terms of miss-detection rates. The FEDD_$cos$ and FEDD_$pear$ presented up to 33% of miss-detection rates in 3/4 of the the time series with abrupt drifts, which means the miss-detection of one of the three known concept drifts. In the time series with gradual drifts, the FEDD_$pear$ reduced the miss-detection rates to 0 in almost all time series.

The second objective of the experiments is to compare FEDD with other drift detection tests based on ELM prediction error. In order to do so, we compare FEDD_$cos$ and FEDD_$pear$ with ELM_DDM and ELM_PHt. In terms of false alarms, the Friedman test presented a $p$-value of $6.6391e$-134, which rejects the null hypothesis of equivalence among the evaluated methods. The Nemenyi test (Figure 3(a)) showed that FEDD_$cos$ and FEDD_$pear$ presented the best results. ELM_PHt presented a low rank, but statistically different from the best method. The box-plots (Figure 4(a)) show that ELM_PHt has a good performance in terms of false alarms compared with the other ELM-based drift detection methods.

In terms of drift detection delay, the Friedman test presented a $p$-value of $6.52126e$-83, which indicates statistical differences. The Nemenyi test (Figure 3(b)) shows that FEDD_$pear$ outperform all the other drift detection methods. The box-plots illustrated in Figure 4(b) show that the methods presented higher drift detection delays in time series with gradual drifts when compared to time series with abrupt drifts. The reason for this is the fact that the concept drifts take more instances to be completed.

In the comparison of miss-detection rates, the Friedman test presented a $p$-value of $6.79843e$-108, which indicates no equivalence among the evaluated methods. The Nemenyi test shows no statistical differences between FEDD$_{cos}$, FEDD$_{pear}$ (Figure 3(c)). The test also shows that ELM_DDM presented a low rank. The reason for the lower miss-detection rates of ELM_DDM is because of the high number of false alarms presented by this method. These metrics are negatively correlated, so a method with high number of false alarms has a high probability of identify true drifts, which decreases the miss-detection rates. Figure 4(c) shows the magnitudes of the differences in performance of the methods in terms of miss-detection rates. Due to its bad drift identification performance, ELM_PHt presented lower false alarms, and consequently the worst results in terms of miss-detection.

The third objective of the experiments is to compare the effectiveness of FEDD$_{cos}$ and FEDD$_{pear}$. The Friedman tests showed that these methods presented statistically equivalent results in terms of false alarms and miss-detection rates. In the case of drift detection delays, FEDD_$pear$ presented a significantly better result than FEDD_$cos$ according to Nemenyi test (Figure 3(b)). The reason for this fact may be because the Pearson distance measures the dissimilarity of two vectors as a function of the correlation of these vectors. Any change in a feature causes a sensitive change in the correlation among the feature vectors. The cosine distance, on the other hand, measures the dissimilarity among two vectors as the angle formed by them. In a feature space with many dimensions, a change in a feature does not imply in a sensitive change in the angle. Because of this, the cosine distance increases more slowly and the drift test takes more instances to detect changes.

In summary, the experiments show that FEDD methods have better drift detection accuracy than ELM_ECDD, which validates the hypothesis stated in this work. FEDD also presented a better trade-off between the number of false alarms,

drift detection delay and miss-detection rates than ELM_DDM and ELM_PHt. FEDD_$pear$ can be considered slightly superior to FEDD_$cos$ since it presents lower drift detection delays at the same time it present an equivalent number of false alarms and miss-detection rates.

## VI. Conclusion

This paper presents a new approach to explicitly detect concept drifts in time series in an online way. FEDD is a drift detection method which monitors some statistical features of the time series in order to identify concept drifts. FEDD uses a feature vector as a reference for the known concept and monitors the evolution of this feature vector in order to test the occurrence of concept drifts. Two distance measures, the cosine distance and Pearson correlation distance, were investigated to compute the feature vectors dissimilarities.

In the computational experiments, we compare FEDD with error-based explicit drift detection methods, namely the ELM_ECDD, ELM_DDM and ELM_PHt. We compared the methods in terms of false alarms, delay to drift point and miss-detection rates when applied to linear and nonlinear time series with abrupt and gradual concept drifts. The results showed that the proposed FEDD presented better drift detection accuracy, which validates the hypothesis stated in this work.

There are several ways to go further with this research. One of them is to integrate FEDD with a regression algorithm to build an adaptive forecasting method which is robust to concept drifts. A second interesting further research would be the investigation of other linear and nonlinear time series features that could be used to describe time series concepts in order to improve the identification of concept drifts. A third possible future research is the investigation of other statistical drift detection tests which could be integrated with FEDD in order to improve drift detection accuracy.

## References


[1] P. S. Cowpertwait and A. V. Metcalfe, *Introductory time series with R*. Springer, 2009.

[2] J.-Z. Wang, J.-J. Wang, Z.-G. Zhang, and S.-P. Guo, "Forecasting stock indices with back propagation neural network," *Expert Systems With Aplications*, vol. 38, no. 11, pp. 14 346–14 355, 2011.

[3] D. Kumar and S. Murugan, "Performance analysis of Indian stock market index using neural network time series model," in *Int. Conf. on Pattern Recognition, Informatics and Mobile Engineering*, 2013, pp. 72–78.

[4] C.-J. Lu, T.-S. Lee, and C.-C. Chiu, "Financial time series forecasting using independent component analysis and support vector regression," *Decision Support Systems*, vol. 47, no. 2, pp. 115–125, 2009.

[5] R. C. Cavalcante and A. L. Oliveira, "An approach to handle concept drift in financial time series based on Extreme Learning Machines and explicit Drift Detection," in *International Joint Conference on Neural Networks*. IEEE, 2015, pp. 1–8.

[6] J. Gama, "A survey on learning from data streams: current and future trends," *Progress in Artificial Intelligence*, vol. 1, pp. 45–55, 2012.

[7] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.

[8] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Adaptive models in uncertain environments," *IEEE Computational Intelligence Magazine*, accepted, 2015.

[9] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: An ensemble method for drifting concepts," *The Journal of Machine Learning Research*, vol. 8, pp. 2755–2790, 2007.

[10] J. A. Guajardo, R. Weber, and J. Miranda, "A model updating strategy for predicting time series with seasonal patterns," *Applied Soft Computing*, vol. 10, no. 1, pp. 276–283, 2010.

[11] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *AAI–SBIA*. Springer, 2004, pp. 286–295.

[12] C. Alippi, G. Boracchi, and M. Roveri, "A just-in-time adaptive classification system based on the intersection of confidence intervals rule," *Neural Networks*, vol. 24, no. 8, pp. 791–800, 2011.

[13] G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand, "Exponentially weighted moving average charts for detecting concept drift," *Pattern Recognition Letters*, vol. 33, no. 2, pp. 191–198, 2012.

[14] L. Auret and C. Aldrich, "Change point detection in time series data with random forests," *Control Engineering Practice*, vol. 18, no. 8, pp. 990–1002, 2010.

[15] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," *Neural Networks*, vol. 43, pp. 72–83, 2013.

[16] J. A. Ferreira, R. H. Loschi, and M. A. Costa, "Detecting changes in time series: A product partition model with across-cluster correlation," *Signal Processing*, vol. 96, pp. 212–227, 2014.

[17] A. S. Block, M. B. Righi, S. G. Schlender, and D. A. Coronel, "Investigating dynamic conditional correlation between crude oil and fuels in non-linear framework: The financial and economic role of structural breaks," *Energy Economics*, vol. 49, pp. 23–32, 2015.

[18] S. Gu, Y. Tan, and X. He, "Recentness biased learning for time series forecasting," *Information Sciences*, vol. 237, pp. 29–38, 2013.

[19] C. Alippi, G. Boracchi, and M. Roveri, "Ensembles of change-point methods to estimate the change point in residual sequences," *Soft Computing*, vol. 17, no. 11, pp. 1971–1981, 2013.

[20] G. Boracchi and M. Roveri, "Exploiting self-similarity for change detection," in *International Joint Conference on Neural Networks*. IEEE, 2014, pp. 3339–3346.

[21] R. B. Prudêncio, T. B. Ludermir, and F. d. A. de Carvalho, "A modal symbolic classifier for selecting time series models," *Pattern Recognition Letters*, vol. 25, no. 8, pp. 911–921, 2004.

[22] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering – A decade review," *Information Systems*, vol. 53, pp. 16–38, 2015.

[23] R. B. Prudêncio and T. B. Ludermir, "Meta-learning approaches to selecting time series models," *Neurocomputing*, vol. 61, pp. 121–137, 2004.

[24] D. Kugiumtzis and A. Tsimpiris, "Measures of analysis of time series (MATS): A MATLAB toolkit for computation of multiple measures on time series data bases," *Journal of Statistical Software*, vol. 33, no. 5, 2010.

[25] A. Strehl, J. Ghosh, and R. Mooney, "Impact of similarity measures on web-page clustering," in *Workshop on Artificial Intelligence for Web Search (AAAI 2000)*, 2000, pp. 58–64.

[26] E. Page, "Continuous inspection schemes," *Biometrika*, pp. 100–115, 1954.

[27] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[28] L. L. Minku and X. Yao, "DDD: A new ensemble approach for dealing with concept drift," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 4, pp. 619–633, 2012.

[29] G. P. Zhang, B. E. Patuwo, and M. Y. Hu, "A simulation study of artificial neural networks for nonlinear time-series forecasting," *Computers & Operations Research*, vol. 28, no. 4, pp. 381–396, 2001.

[30] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[31] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[32] P. Nemenyi, "Distribution-free multiple comparisons," in *Biometrics*, vol. 18, no. 2. International Biometric Society, 1962, p. 263.